CREDIT CARD TRANSACTION

dimpybangoriya@gmail.com

INTRODUCTION

Credit card transactions generate vast amounts of data daily, providing valuable insights into customer spending behavior, banking trends, and potential fraudulent activities. This project focuses on analyzing credit card transaction data using SQL to ensure data integrity, extract meaningful business insights, and detect fraudulent activities.

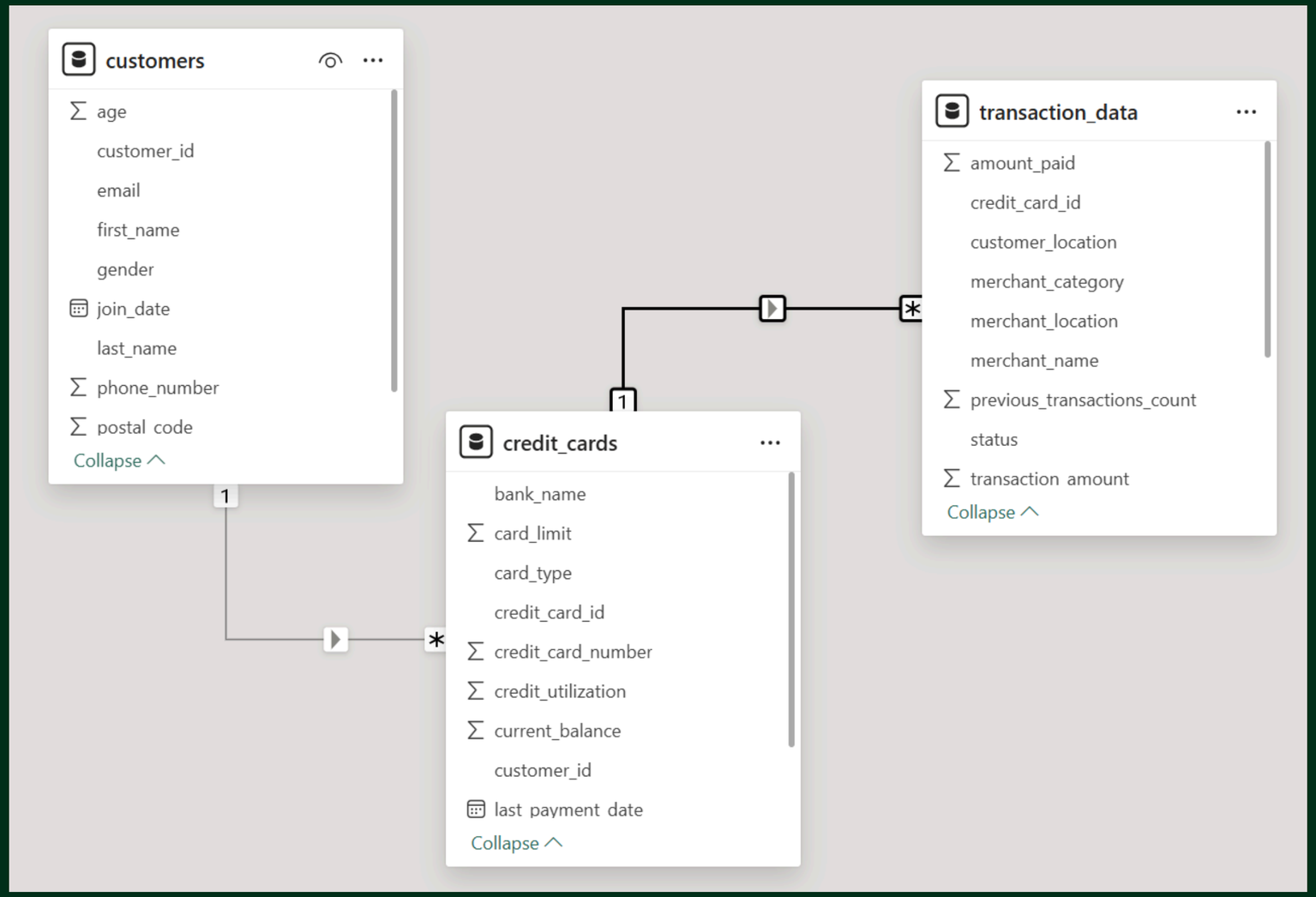
The analysis is structured into three key steps. First, data cleaning is performed to handle missing values, incorrect references, and duplicate records, ensuring a high level of accuracy and reliability. Next, transactional data is analyzed to uncover spending patterns, monthly trends, top customers, and the most commonly used banks. Finally, business insights are derived to help banks enhance customer engagement, optimize service offerings, and improve fraud prevention strategies.

By leveraging SQL for data cleaning and analysis, this project aims to assist financial institutions in understanding customer behavior, enhancing security measures, and making data-driven decisions to improve overall operational efficiency.

Table used

- Customer
- Credit Card
- Transaction Data

ER Diagram


customers
age
customer_id
email
first_name
gender
join_date
last_name
phone_number
postal code
Collapse
1
credit_cards
bank_name
card_limit
card_type
credit_card_id
credit_card_number
credit_utilization
current_balance
customer_id
last payment date
Collapse
1
transaction_data
amount_paid
credit_card_id
customer_location
merchant_category
merchant_location
merchant_name
previous_transactions_count
status
transaction amount
Collapse

Creating Table

```sql
-- ----------customers--------

create table bank.customers (
customer_id int(100) Primary Key,
first_name varchar(100),
last_name varchar(100),
email varchar(100),
phone_no varchar(100),
state varchar(100),
postal_code varchar(100),
age int(50),
gender varchar(50),
join_date date
```

```sql
create table bank.customers (
customer_id int(100) Primary Key,
first_name varchar(100),
last_name varchar(100),
email varchar(100),
phone_no varchar(100),
state varchar(100),
postal_code varchar(100),
age int(50),
gender varchar(50),
join_date date

);
```

```sql
-- ----------transaction detail--------

create table bank.transaction_detail(
transaction_id int(100) Primary Key,
credit_card_id int(100),
transaction_date varchar(200),
transaction_time varchar(200),
transaction_amount varchar(300),
transaction_category varchar(200),
customer_location varchar(200),
merchant_name varchar(200),
merchant_location varchar(200),
transaction_mode varchar(200),
amount_paid varchar(300),
previous_transactions_count varchar(200),
merchant_category varchar(200),
status_detail varchar(150),
Foreign Key(credit_card_id) References credit_card(credit_card_id)
);
```

Checking For Missing Data

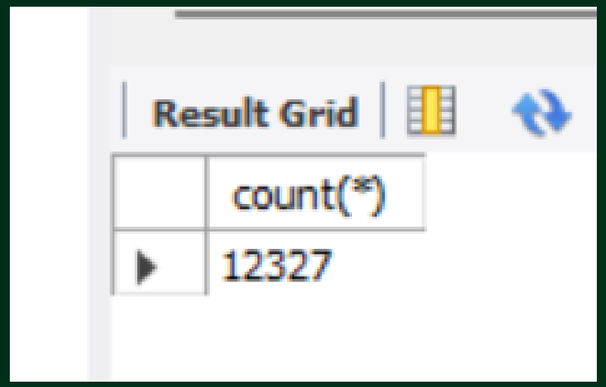
```sql
-- Checking for Missing Data:

	--  Find transactions where credit_card_id is missing.

	select *
	  from bank.transaction_detail t1
	  where t1.credit_card_id is Null;

	 -- Identify credit cards that reference a non-existing customer.

	 SELECT *
	 FROM credit_card cc
	 LEFT JOIN customers c ON cc.customer_id = c.customer_id
	 WHERE c.customer_id IS NULL;
```

Removing Invalid Data

```sql
--  Removing Invalid Data:

        --  Delete transactions with a missing or incorrect credit_card_id.

                select *  FROM transaction_detail
WHERE credit_card_id IS NULL OR credit_card_id NOT IN (SELECT credit_card_id FROM credit_card);


        --  Remove credit cards linked to customers who don't exist in the customer table.

            select *
            from credit_card where customer_id Not in (select customer_id from customers);

            SELECT cc.* FROM credit_card cc
            LEFT JOIN customers c ON cc.customer_id = c.customer_id
            WHERE c.customer_id IS NULL;
```

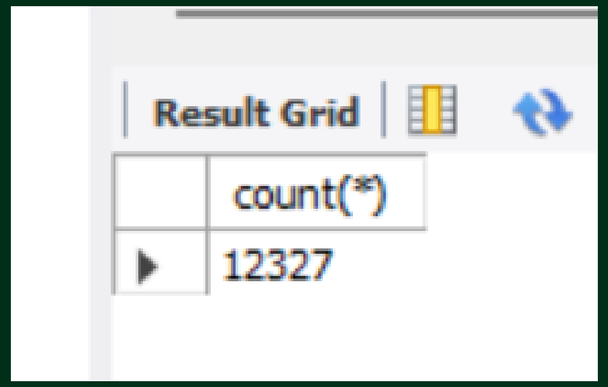
Total Transaction

```sql
-- Total Number Of Transaction

select count(*) from
transaction_detail;
```



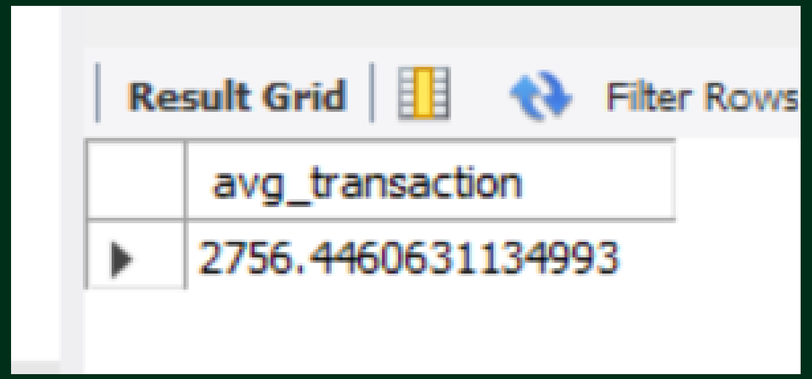
Total Amount Spent

```sql
-- What is the total amount spent?

SELECT SUM(transaction_amount) AS total_spent
FROM transaction_detail;
```



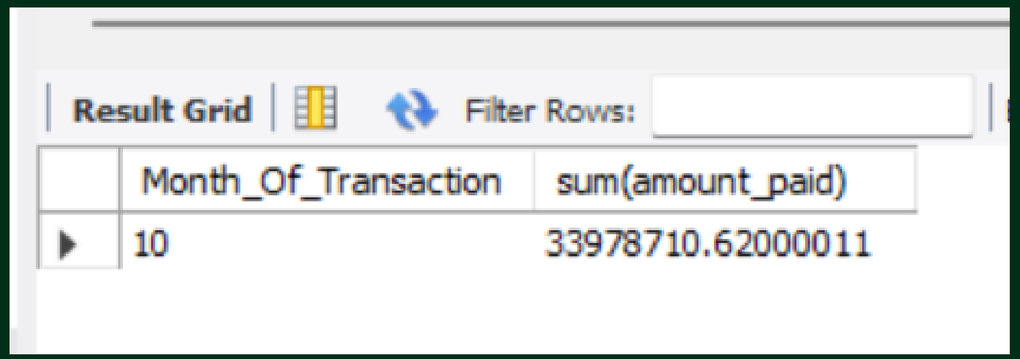
Average Transaction

```sql
-- Avg Transaction Spent

SELECT AVG(transaction_amount) AS avg_transaction
FROM transaction_detail;
```



Variation Of Spending By Month

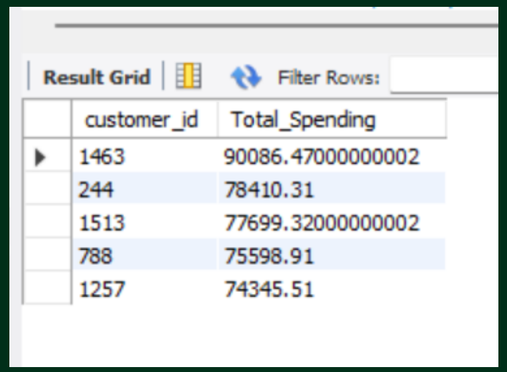
```sql
--  How does spending change month by month?

select *,MONTH(transaction_date) as Month_Of_Transaction
from bank.transaction_detail t1;

Select x.Month_Of_Transaction,sum(amount_paid)
from (select *,MONTH(transaction_date) as Month_Of_Transaction
from bank.transaction_detail t1)x
group by Month_Of_Transaction;
```



Month_Of_Transaction	sum(amount_paid)
10	33978710.62000011

Top 5 Customers By Total Spending

```sql
-- Top 5 customers by total spending

select c1.customer_id,sum(t1.amount_paid) as Total_Spending
from  bank.customers c1
Left Join bank.credit_card cc1
on c1.customer_id=cc1.customer_id
Left Join bank.transaction_detail t1
on cc1.credit_card_id=t1.credit_card_id
group by c1.customer_id
order by Total_Spending Desc
Limit 5;
```

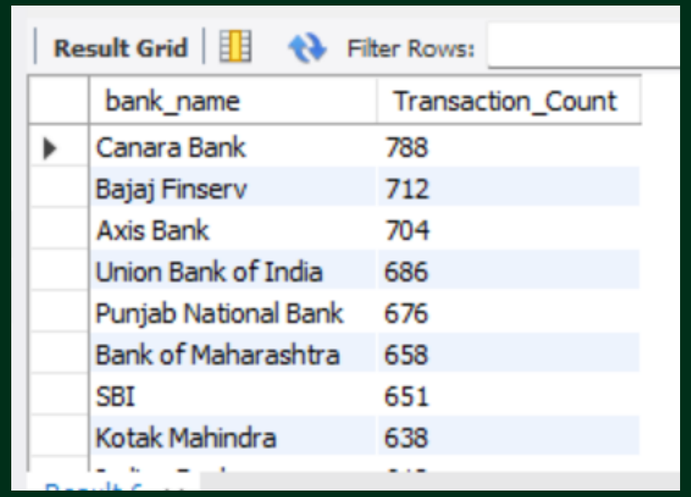


Result Grid | Filter Rows:

customer_id	Total_Spending
1463	90086.47000000002
244	78410.31
1513	77699.32000000002
788	75598.91
1257	74345.51

Frequently Used Credit Card

```sql
-- Most frequently used credit cards by bank

select cc1.bank_name,count(t1.transaction_id) as Transaction_Count
from bank.transaction_detail t1
Left Join bank.credit_card cc1
on t1.credit_card_id=cc1.credit_card_id
group by cc1.bank_name
order by Transaction_Count Desc;
```



Result Grid | Filter Rows:

bank_name	Transaction_Count
Canara Bank	788
Bajaj Finserv	712
Axis Bank	704
Union Bank of India	686
Punjab National Bank	676
Bank of Maharashtra	658
SBI	651
Kotak Mahindra	638

Customer With Highest Avg. Transaction Value

```sql
-- Customers with highest average transaction value


select c1.customer_id,avg(t1.transaction_amount) as Average_Transaction_Amount
from bank.customers c1
Left Join bank.credit_card cc1
on c1.customer_id=cc1.customer_id
Left Join bank.transaction_detail t1
on cc1.credit_card_id=t1.credit_card_id
group by c1.customer_id
Order by Average_Transaction_Amount Desc;
```

Result Grid | Filter Rows:

bank_name	Transaction_Count
Canara Bank	788
Bajaj Finserv	712
Axis Bank	704
Union Bank of India	686
Punjab National Bank	676
Bank of Maharashtra	658
SBI	651
Kotak Mahindra	638

Top 5 Banks By Total Transaction Value

```sql
-- Top 5 banks by total transaction value

select cc1.bank_name,sum(t1.transaction_amount) as Total_Transaction_Value
from bank.transaction_detail t1
Left Join bank.credit_card cc1
on t1.credit_card_id=cc1.credit_card_id
group by cc1.bank_name
order by Total_Transaction_Value Desc
Limit 5;
```

157 from bank.transaction_detail t1

Result Grid | Filter Rows: | Export:

bank_name	Total_Transaction_Value
Canara Bank	2171064.9900000016
Bajaj Finserv	1953746.380000001
Axis Bank	1911830.3899999985
Union Bank of India	1880907.76
Bank of Maharashtra	1876609.5700000003

CONCLUSION

This project successfully analyzes credit card transactions using SQL, focusing on data integrity, business insights, and fraud detection. By cleaning the data, we ensure accuracy and reliability for analysis. The transaction trends, customer spending patterns, and bank performance insights provide valuable information for decision-making. These findings help banks enhance customer service, target high-value customers, and improve fraud prevention measures, making credit card usage more secure and efficient.

THANK YOU

dimpybangoriya@gmail.com